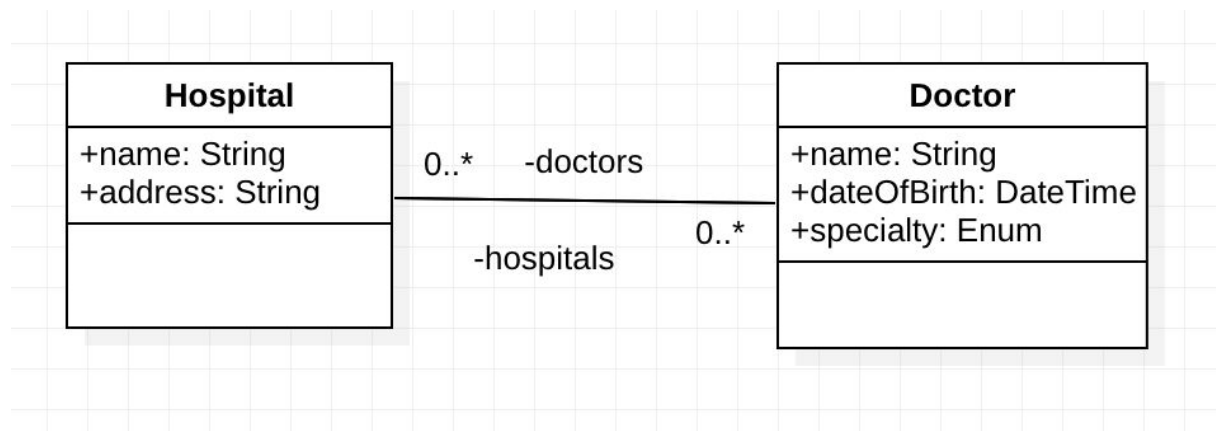


Última prueba de actuación - Diseño de Aplicaciones 1

Hora de inicio: 19:55 Hora de fin: 21:00

Ejercicio

Dado el siguiente diagrama de clases



Primera parte

Implemente utilizando Entity Framework las operaciones de **crear, actualizar, borrar y obtener** un *Hospital* y un *Doctor* teniendo en cuenta que

- El nombre del *Hospital* es requerido
- El nombre del *Doctor* es requerido
- Un *Hospital* puede tener o no *Doctors*
- Un *Doctor* puede o no trabajar en *Hospitals*
- El atributo *specialty* del *Doctor* es un enumerado con valores “Pediatria”, “Traumatología” y “Fisiatria”

Para las restricciones deberá utilizar Data Annotations ó Fluent Api.

Segunda parte

Implemente utilizando **LINQ to queries** las siguientes operaciones

- Obtener el doctor más viejo que trabaje en un hospital
- Obtener los doctores, que trabajen o no, con especialidad *Pediatria* ó *Fisiatria*

Aclaraciones

- No es necesario que haga tests
- Puede codificar todo en un único proyecto, utilizando una aplicación por consola
- Si le facilita puede utilizar *eager loading* para el manejo de relaciones
- Puede o no utilizar migrations
- Será **fuertemente** valorada la calidad del código

Formato de la entrega

Entregar vía correo electrónico un link al repositorio en Github con la solución del ejercicio. El repositorio puede ser público, de ser privado se deberá dar el acceso correspondiente.

Se deberá generar un *tag* release a la branch *master* del repositorio. Desde este release se corregirá el ejercicio.

<https://help.github.com/en/articles/creating-releases>